



SYSTEMS ENGINEERING NEWSLETTER

brought to you by

Project Performance International (PPI)

SyEN 57 – September 18, 2017

SyEN is an independent free newsletter containing informative reading for the technical project professional, with scores of news and other items summarizing developments in the field, including related industry, month by month. This newsletter and a newsletter archive are also available at www.ppi-int.com.

Systems engineering can be thought of as the problem-independent, solution technology-independent, life-cycle-oriented principles and methods, based on systems thinking, for defining, performing, and controlling the engineering effort within a technical project. The approach aims to maximize the benefit delivered to the enterprise, as influenced by the needs and values of the applicable stakeholders.

If you are presently receiving this newsletter from an associate, you may wish to receive the newsletter directly in future by signing up for this free service of PPI, using the form at www.ppi-int.com. If you do not wish to receive future Systems Engineering Newsletters, please unsubscribe by clicking on the link at the bottom of this email.

We hope that you find this newsletter to be informative and useful. Please tell us what you think. Email us at syen@ppi-int.info.

The views expressed in externally authored articles are those of the author(s), and not necessarily those of PPI or its professional staff.

IN THIS EDITION

A Quotation to Open On

[Read More...](#)

Feature Article

- The Impact of Effective Agile Teams on Organizations, *by Rainer Grau*

[Read More...](#)

Article

- Beyond the “Vee” Model: The Wedge Model™, *by Robert J. Halligan*
- Integrating Program Management and Systems Engineering, *by Dr. Ralph Young*

[Read More...](#)

Systems Engineering News

- INCOSE Washington Metropolitan Area Chapter to Sponsor a STEM Initiative
- Software Brittleness May Harden Embedded Systems
- INCOSE-PMI Collaboration
- Nominations are Open for The Lt Gen Thomas R. Ferguson, Jr. Systems Engineering Excellence Award Program
- INCOSE Webinar: Agile Systems & Processes 106: Risk Management and Mitigation
- SERC Talks: What are the Top Ten Software Security Flaws?

[Read More...](#)

Featured Organizations

- The International Systems Engineering Network

[Read More...](#)

Conferences and Meetings

[Read More...](#)

Some Systems Engineering-Relevant Websites

[Read More...](#)

Systems Engineering Publications

- How Systems Engineering Can Help Fix Health Care
- Requirements Engineering
- “Suits you sir! – Choosing the Right Style of SE before Tailoring to Fit” Using Functional Failure Modes and Effects Analysis to Guide Selection of the Right Systems Approach
- Navigating Complexity: A Practice Guide
- Journal of Systems and Software on Quality Engineering and Management of Software-Intensive Systems (Special Issue)

[Read More...](#)

Systems Engineering Tools News

- Simulation Boom Stalls. CAE Experts Needed to Make Engineering Apps
- Charles River Analytics Building Tools for Decision-making under Uncertain Conditions
- Dassault Systèmes and No Magic Establish Partnership to Boost Systems Engineering Solutions
- Jama

[Read More...](#)

Education and Academia

- Architecture and Systems Engineering: Models and Methods to Manage Complex Systems
- INCOSE Institute for Technical Leadership

- New Face, New Course Coming to Colorado State University Systems Engineering Program
- University of Alabama Huntsville (USA) Industrial and System Engineering Students Build Successful Business Model for Local Medical Practice
- Collaborate Through Stories: Using Interactive Storytelling to Bridge the Gap between Engineers and Stakeholders

[Read More...](#)

Standards and Guides

- The International Standards Organization

[Read More...](#)

Some Definitions to Close On

- A Glossary of Systems Engineering Terms from the Guide to the Systems Engineering Body of Knowledge (SEBoK) v. 1.8
- The Internet of Things Explained

[Read More...](#)

PPI and CTI News

[Read More...](#)

Upcoming PPI and CTI Participation in Professional Conferences

[Read More...](#)

PPI and CTI Events

[Read More...](#)

A QUOTATION TO OPEN ON

“When compared to typical systems engineering endeavors, the application of model-based systems engineering delivers a 55% reduction of the total development cost. The most mature form of MBSE is model-based Product Line Engineering (PLE). When compared to typical systems engineering endeavors, the application of model-based product line engineering delivers a 41.6% reduction of the total development cost.”

Jerry Krasner, “How Product Development Organizations can Achieve Long-Term Cost Savings Using Model-Based Systems Engineering”, White Paper

FEATURE ARTICLE

The Impact of Effective Agile Teams on Organizations

by

Rainer Grau

DigitecGalaxus AG

Version 2.0

June 21, 2017

Abstract

Agility is a recognized success factor of companies. Nevertheless, many companies struggle to establish agility in the whole organization. A core aspect of the transformation toward an effective agile organization is management itself. The transformation towards agility is more than a change of processes, methods, and techniques. Different concepts in management are required to complete the transformation. This article discusses aspects in management that are critical in an agile transformation. It presents options of potential management models with a focus on delegation of management tasks to agile teams, new concepts of management such as part-time managers, and new concepts of performance evaluation.

Email: rg@juopera.com

Introduction

This article discusses the evolution and transformation of work organizations from classical hierarchical organizations to agile organizations. Whereas classical organizations build teams around projects, in today's agile organizations work flows continuously into stable teams following typically some kind of Kanban¹ implementation.

This change in work flow and team organization requires a change in the management and leadership model as well. Strategy and goal setting processes change, performance evaluation changes from an individual evaluation to both a team and individual evaluation, and a set of management activities including planning and decentralized control moves from managers into teams. The budgeting process changes from annual budgeting to a rolling forecast system.

This article addresses two aspects of this change. The first aspect is the change of the work organization in agile organizations. The other aspect is the required change of the management and leadership model to form a vision, derive a roadmap, and orchestrate an organization of aligned agile teams that comprise the strategy process and are responsible for implementation.

History of Work Organization

Looking back to the work organizations of the 1980's, software development was (astonishingly) not that different from today's agile organizations. The main difference is that software development was a profession of very few individuals that owned the full skill set to translate business needs into an implemented, functional system. The complexity of environments, programming languages and options to develop software was very low. Additionally, no education paths existed to learn software development as a profession on its own. So-called "programmers" were recognized as strange people doing magic things with expensive and huge equipment called "computers". From today's point of view, the first generation of software engineers integrated all required disciplines from business process development to requirements engineering to coding, testing, deployment, and maintenance.

As systems started to become increasingly complex, and communicated over networks with each other, the first-generation "programmers" failed in many well-known situations where software development costs suddenly increased dramatically. We recall that in these early days, hardware was extremely expensive and thus a critical component in determining potential solutions for business needs. Software was included in the cost of the hardware. Some of the reasons for these failures of the day included an absence of knowledge concerning system and software architecture, structured methods and techniques to establish

¹Kanban is a method for managing knowledge work with an emphasis on just-in-time delivery while not overloading the team members. This approach presents all participants with a full view of the process from task definition to delivery to a customer. Team members pull work from a queue.

a software life cycle, and the support of the software development lifecycle (automation and collaboration tools).

The response to these failures was the development of structured processes, methods, and techniques. Some of the most commonly used process models were the Structured Systems Analysis and Design Method (SSADM)², the Vee Model, and the Rational Unified Process (RUP). The classical life cycle models for software development were treated as waterfall processes. For example, SSADM explicitly recommended going through the lifecycle in short iterations in order to learn more and more about system needs as one proceeded. In the early 1990's, Barry Boehm evolved the spiral model, an approach that is not unlike iterative models that are popular today.

The interesting question is: why did these process models fail? In the opinion of the author, two main factors had far more influence than the process models themselves: the underlying management and leadership model and specialization of job profiles.

The Classical Role of Line management

Personnel in companies were divided into three distinct groups:

1. **Business Development:** The members in this group knew the market and were responsible for business development and business operations. In addition to executive management, sales or marketing employees were found in this group. The critical role of this group was to decide what projects would be funded. This group of people was responsible for the profitability of the company. They determined the budgeting. In summary, the people in this group provided the intellectual capability and served as the investment authority of the company.
2. **Middle Management:** The members of this group were responsible to execute the funded projects using project management techniques, i.e. by task assignment, controlling, tracking and reporting to specialists in the work force that executed the assigned tasks. Members of middle management were often referred to as the "sandwich managers", since they were assigned goals set by the members of the business development group and experienced difficulty as a result of the project teams continually requesting better support, tools, environments, and education.
3. **Specialists:** The members of this group of people were highly educated persons in one or more disciplines required to implement a complex system. These disciplines evolved into job profiles as well: business analyst, requirements engineer, software developer, tester, operations manager, and many others. The specialists executed the assigned tasks of middle management to reach the goals set by the business developers. Specialists often had been organized as pools of X (where X=any discipline) with a middle manager as the line manager of this pool.

² Structured Systems Analysis and Design Method (SSADM), originally released as methodology, is a systems approach to the analysis and design of information systems.

If we consider the typical line management organization in today's companies, we recognize two distinct line management hierarchies. The first one is the business management hierarchy, starting from the Chief Executive Officer (CEO), including the Chief Financial Officer (CFO) and often the head of sales, the head of marketing, and the head of business development, each with its own specific sub-hierarchies. This business management hierarchy had the real power in the organization since these people decided about financial matters.

The second management hierarchy is the development organization, often with the Chief Information Officer (CIO) as top-level manager, complemented by the head of project management, the heads of areas such as business analysis, software development, testing, and process & tools. This development management hierarchy had low power in the organization because these persons, including the CIO, had strict constraints in the form of budgets and personal performance objectives. Typically, the development organization was defined as a cost center and not as part of the value stream of the organization.

The structural failure in this type of organization is the distributed responsibility. The weakness is the lack of authority and empowerment that would enable the development organization to contribute to the organization's value stream. The business development group did not feel responsible for the project results, especially for the quality of the outcomes. The middle management group did not feel responsible for the strategy on the one hand, nor for the outcomes on the other hand. The group of specialists did not feel responsible for the overall company goals; rather only for the fulfilment of specific project objectives, which may not even have been aligned to the strategy.

Mindset Change in Agile Organizations

The core mindset change in effective agile organizations is the holistic responsibility of teams. To understand this statement, the two terms "effective agile organization" and "holistic responsibility" need to be explained.

Within an effective agile organization, agility is not restricted to the development organization. Rather, *all* employees participate, starting with the CEO down to data entry personal and retail sales persons on the shop floor. The organization is aligned along the value streams in the company. *As a prerequisite, the value streams must be known and transparent.* Within an effective agile organization, one or more agile teams own a very specific part of a value stream with all aspects. The activities of an agile team create an outcome that realizes a business value for the company. For example, two customer service teams "own" the email service responsibility. They work closely with a development team that implements needs and requirements of the two customer service teams through the software tools used by the customer service teams.

This is where "holistic responsibility" comes into play. In an effective agile organization, the core responsibility of the executive team is to optimally arrange teams along the value streams, set motivating benchmarks of strategic goals, and provide clear guidelines for an inter-team collaboration. For example, the executive team could provide the two customer service teams the benchmark indicators of customer

satisfaction of the main competitors and the motivating idea to excel the competition. Agile teams then derive their specific goals from such benchmarks, identify the required measures to reach these self-set goals, and manage the realization of the measures within the teams. This mindset of action is the definition for “holistic responsibility”. Operation, development, and management of all activities is within the teams. The teams own all skills to optimize themselves and find their optimal team organization under the given constraints of the overall organization.

One might ask, what is the impact of having an effective agile organization, with teams taking holistic responsibility for a part of a company’s value stream?

Instead of having the three groups noted above (business development, middle management, and specialists), an agile company is built as follows:

- An executive team drives strategy, builds agile teams along the value streams of the company, and sets guidelines for collaboration of the agile teams. The executive team defines and explains transparent and motivating benchmarks to the teams, so they can derive self-set goals. The executive team supports the agile teams to take responsibility and to develop all skills required to fulfil the holistic responsibility for its part of the value stream.
- The agile teams take responsibility for a part of a company’s value stream. One or more agile teams typically work very closely together (high cohesion) to realize the owned part of the value stream. The teams derive goals, measures, and concrete actions from the given benchmarks. They give feedback to the executive team about the feasibility of the benchmarks, the required actions, and missing skills. A core mindset of an agile team is the intrinsic motivation to realize business value for the company and also to continuously improve themselves to create the business value with less input (this perspective may be characterized as a “lean mindset”).

The main difference from the classical organization is that many aspects of management tasks are transferred and delegated to the agile teams. *Typically, every employee executes management activities as well.* Scrum as one example of an agile team arrangement visualizes this very clearly: Planning, estimation, decomposition of business requirements into technical implementation tasks are all performed by the team members, not by an additional manager.

It is obvious that organizing as an effective agile company with holistic responsibility given to teams has a significant impact on line management. One may wonder whether management is even needed!

Deployment of Agile Work Organization

It is interesting to consider an example of an agile process methodology in terms of the management aspect. “Scrum”³ is a good template to consider because it implements interesting concepts that can be abstracted, scaled, and transformed into agile organizations.

In Scrum, there are two roles, “Product Owner” and “Scrum Master”. Neither of these roles is solely devoted to “management”. The Product Owner is responsible for creation of business value; the Scrum Master is responsible to embed the team into the guidelines for the organization and to support and serve the team in providing a clean and continuously optimizing Scrum implementation. In addition to their “management” responsibilities, the Product Owner and the Scrum Master execute development tasks such as analysis, requirements engineering, or coding – or whatever the team needs in order to perform its responsibilities to contribute to the organization’s value stream.

The Product Owner role and the Scrum Master role are good templates to “abstract”. The abstraction of a Product Owner leads to a group of persons that feel responsible for business value and transport this responsibility into the teams so that every team member participates in this responsibility. In scaling agility frameworks, different terms exist for such individuals: Chief Product Owner, Epic Owner, Business Process Owner, for example. Overcoming the confusion of terminology and scaling frameworks, the common sense approach assures that all team members feel responsible for business value, and execute appropriate operational tasks accordingly.

The abstraction of a Scrum Master leads to a group of persons that feel responsible to identify collaboration rules for teams and – combined with these collaboration rules – the constraints a team is acting under. As well, they feel responsible to develop and educate individuals and processes, methods, disciplines, and tools so that a team is as autonomous as possible and owns all the skills, tools, and support needed to create the business value. In scaling agility frameworks, different terms exist for such individuals as well, for example, Agile Agent or Release Train Engineer. Again, overcoming the confusion of terminology and

³ **Scrum** is an iterative and incremental [agile software development](#) framework for managing product development. It defines "a flexible, [holistic](#) product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach" to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines involved. A key principle of Scrum is the dual recognition that customers will change their minds about what they want or need (often called requirements volatility) and that there will be unpredictable challenges—for which a predictive or planned approach is not suited. As such, Scrum adopts an evidence-based empirical approach—accepting that the problem cannot be fully understood or defined up front, and instead focusing on how to maximize the team's ability to deliver quickly, to respond to emerging requirements, and to adapt to evolving technologies and changes in market conditions.

scaling frameworks, the common sense approach assures that all team members feel responsible for business value, and execute appropriate operational tasks accordingly.

The Impact of Effective Agile Teams on Organizations

There are two orthogonal aspects that are key to building an agile organization: 1) There are individuals feeling responsible for business value creation; and 2) there are individuals feeling responsible for the development of individuals and the organization. None of these individuals are full-time managers. As full-time managers, they would lose contact to the base and would not be able to take responsibility. To be part of the agile team is a core success factor.

This view on agile organizations answers the question whether managers are still required. The answer is, clearly “yes”. Individuals own management responsibilities; however, these individuals are not full-time managers. The extent of effort that these part-time managers act in management activities depends on the responsibility and the status of the organization. Additionally, as discussed earlier in this article, every employee participates to some degree in management activities. In the final analysis, the part-time managers in agile organizations own roles with special and well understood responsibilities, either for business value or for development of individuals and organization.

One might ask, “What about the CEO? Is the CEO in an agile organization as well a role and a part-time manager?” This question is not answered yet. Based on personal observations of the author in different companies, no clear answer is given and no clear answer is needed. An organization can even define this as an attribute of the organization. *In more classically aligned companies or companies in the middle of transformation from classical to agile organizations, full-time management still is required.* The responsibility of this full-time management is to define strategy, set benchmarks, and derive goals for the teams if the teams are not yet enabled to act autonomously and be self-contained under the desired guidelines. Full-time managers might be required to develop the organization with the support of additional part-time managers as described above.

Companies that have implemented a “holacracy”⁴ have already crossed that chasm. Management activities are executed in circles that take responsibility for specific aspects of an organization. The individuals in these circles naturally are part-time managers and performers at the same time. The author knows of companies that implemented a holacracy organization where team members took the full responsibility as official members of the steering board for an elective period, supported by experienced staff members.

⁴ **Holacracy** is a social technology or system of organizational governance in which authority and decision-making are distributed throughout a holacracy of self-organizing teams rather than being vested in a management hierarchy. It has been adopted in for-profit and non-profit organizations in the U.S., France, Germany, Switzerland, New Zealand, Australia, and the UK.

Every company must define its own position on an “*agile slider*” between pure classical to true agile or a holacracy organization. There are many flavors, options, and organization forms possible. Each organization must find its optimal setup. The author’s opinion is that companies with the organizational slider on the agile side will outperform the competition in the long run as compared to companies with the slider on the classical side.

Now to answer the question: Is the CEO in an agile organization as well a role and a part-time manager? This is a valid role model and it is already established in organizations. The author works in a company where the CEO takes active part in the development of company features, performing in the roles of business analyst and requirements engineer. The author himself is one of the part-time managers responsible for the development of individuals and the organization, but is as well taking active part in the work activities of the team.

Performance Evaluation in Agile Organizations

Prior to discussing providing line management in agile organizations, it is interesting to consider performance evaluation. The reason is that in classical organizations, performance evaluation is the responsibility of the line managers in a top-down approach. The CEO defines goals for direct reports and evaluates performance against these goals. This construct is deployed throughout the organization. All performance evaluation is typically based on the fulfilment of individual goals.

In an agile organization where agile teams create business value for a section of the value stream of the company, evaluation of the created business value is a key construct of performance evaluation. As stated above, the holistic responsibility for the business value is on the team. Either the team succeeded in reaching the (self-set) goals based on transparent benchmarks, or the team failed. No single individual is responsible for success or failure. This team success or failure is one part of performance evaluation.

The other part of performance evaluation is the personal part in the form of professional behavior and skills. Agile team members have a T-profile, i.e. they are professionals in more than one discipline with a focus on two or more disciplines. The individual performance evaluation is based on the skill level(s) within the T-profile of the individual. Appropriate aspects of the evaluation are, for example, professional level and quality of work results, skill level based on years of experience, potential for future development of skills, and the degree of intrinsic motivation for personal development.

The performance evaluation for any employee is based on the two orthogonal aspects: Team performance based on created business value and individual performance based on skill level, quality of work, motivation, and engagement.

Responsibility of Line management in Agile Organizations

One interesting point is: Who in the company is collecting the two orthogonal performance elements (team performance and individual performance) for every employee and who is responsible to evaluate the performance of an individual based in these performance elements. Is there a need for some form of a

special line manager position that collects feedback for both orthogonal aspects for a set of employees and how would that position fit within an agile organization?

Because widespread experience and research results with true agile organizations are not yet available, it is not possible to assert recommendations. What this article can offer are options and alternatives for organizations to provide a starting point for experimentation and to evolve toward a specific optimal solution for a line management setup. The author offers three alternatives, as follows:

- The “as-is” solution: line management is unchanged when looking on the organigram (organization chart), but responsibilities of the line management are partially delegated.
- The “agile management” solution: The performance evaluation is under the responsibility of (part-time) managers following the model of the part-time managers responsible for creation of business value or organizational development.
- The “holacracy” solution: line management is a part-time responsibility of a group of persons (a circle?) that feels responsible to care about performance evaluation.

These three alternatives consider the extent of agility in an organization. Classical organizations might start or prefer the first alternative. Very agile organizations or holacracies might start with the holacracy solution. Any other organization on an agile level in between might prefer the “agile management” solution.

The “as-is” Solution

The as-is solution describes an alternative for companies starting with company-wide agility. The management line hierarchies are well-established and typically there is no need nor pressure perceived for a change of line management. *The recommendation is, first change the work organization and establish an agile mindset before changing the line management structure.* This assures more stability in the change process. A necessary prerequisite is the support of change by the existing management on all hierarchy levels.

A recommendation for the existing line management is to re-organize team work by building stable teams that create business value for a specific part of the value stream of the organization – not only in software development but for all departments in the company. An example is given above where two (classical) customer service teams and a (classical) software development team re-organize as a unit responsible for customer service. This will create new two or three multidisciplinary teams including customer service agents, business analysts, developers, and whomever else is needed to address all business and development aspects.

In this alternative, the line manager moves certain aspects of management responsibility to the teams. The following management aspects are options to be moved:

- Micro management aspects such as controlling, tracking and reporting are moved to the teams. Stable teams know best how to control, track, and report concerning team results.

- Line managers are responsible that the teams have all of the skills and experience required to act autonomously and successfully.
- Line managers are responsible to carry the vision, targets, and benchmarks for the teams into the teams and support the teams.
- Line managers define team goals together with the teams and collect input for individual performance evaluation concerning skills and engagement of the team members.

These changes are demanding and challenging for line management. *The skill set of line managers' transitions from classical management based on a track and control mindset to being a supporting advisor and providing leadership.*

The most critical change is that line managers are responsible for *teams* on one hand and for individuals on the other hand – direct reports of a line manager act in a team where the team goals are defined by another line manager. In the customer service example, the customer service line manager now defines the benchmarks, targets, and goals of all “new” teams. Nevertheless, the software engineers, business analysts, and others are still direct reports of a manager from the development (line) organization. This is demanding and challenging for a manager because the manager is now communicating with individuals with different education, unique profiles, and individual mindsets.

The “agile management” Solution

In the agile management solution, the change to agile teams aligns to a part of the value stream already in place. Two new types of managers are now in place: the part time manager for business value and the part time manager for individual and operational development.

Reviewing existing scaling agile frameworks reveals that recommendations already exist. The business oriented managers build one part of the line management hierarchy. As agile teams are aligned along the value streams, a complete values stream is the summary of a set of organizational units that must collaborate. *The primary difference from classical organizations is that operational execution and development of the organizational execution are now integrated.*

The managers responsible for individual and operational development build the second part of the line management hierarchy in the form of a “Community of Practice”⁵ (CoP) or “Guild”. These are

⁵ A **community of practice (CoP)** is a group of people who share a concern or a passion for something they do, and learn how to do it better as they interact regularly. The concept was first proposed by cognitive anthropologists Jean Lave and Etienne Wenger in 1991. A CoP can evolve naturally because of the members' common interest in a particular domain or area, or it can be created deliberately with the goal of gaining knowledge related to a specific field.

organizational structures responsible to build up skill and experience in the disciplines required in the agile teams.

One recommendation of frameworks for scaling agility such as Large Enterprise Scaled Scrum (LeSS) is that the heads of the Communities of Practice are responsible to collect all feedback required for performance evaluation. The sources of feedback are the *team performance* coming from the teams (or the business oriented managers responsible for the teams) and – ideally – some form of 360-degree feedback from employees. A second recommendation is that performance evaluation changes toward a more continuous process instead of a yearly goals check. This change fits within the change from a yearly budgeting process to a rolling forecast in financial planning.

Based on the experience of the author, the recommendations of the scaling agility framework need not be followed strictly. There are examples in the companies where business-oriented managers as well are responsible for performance evaluation and managers for individual and operational development take no responsibility for line management. From the experience of the author, core success factors in the agile management solution are the following:

- Management responsibility is transferred and delegated to employees. This enriches every job profile and fosters intrinsic motivation. *The author estimates that about 50% of the required management capacity is moved to teams and can be automated by smart teams.*
- Managers become part-time managers. This aligns management with vision and the company's goals as managers get "their hands dirty". *Additionally, the often-discussed gap between manager and employee may become smaller or even disappear. Managers are to some extent performers.*

The main responsibilities of the executive management in an agile management solution are:

- *to drive strategy and transport strategy and vision into the teams;*
- *to identify the values streams of the company;*
- *to build and align teams along the values streams to optimal design value streams; and*
- *to define, based on feedback from the teams, rules and guidelines for collaboration of teams.*

The "holacracy" Solution

Holacracy is a very radical evolution of organizations toward the distinction between the purpose of the company and the personal development of an individual as team member acting in specific roles. In a holacracy, the most important aspect is the definition of guidelines and rules concerning how teams collaborate; and how new teams are created or existing teams closed. Teams form along the values stream or to cover an organizational need. An individual can be part of more than one team. For example, an employee can be team member of a customer service team and team member of a team that defines guidelines and rules for performance evaluation and salary policies.

A more complete discussion of holacracy is out-of-scope for this article; however the author recommends the existing literature. The author himself does not have any experience with holacracy. The existing experience and examples offer two-fold and ambivalent impressions.

Summary

Agility is more than a set of methods, techniques and rearrangements of processes. The transformation towards an effective agile company requires rethinking the organization of management. Delegation of a set of management responsibilities to employees and building teams that create tangible business value are critical. This approach fosters intrinsic motivation at all levels.

The delegation of management activities to teams provides the opportunity to act as part-time manager either responsible to create business value or responsible for individual and organizational development.

Performance evaluation is based on collection of feedback from orthogonal sources: the outcome of the team and the individual development and engagement. The performance evaluation changes from a yearly activity to a continuous feedback process. It is recommended that the performance evaluation be executed by the managers responsible for individual development and engagement. An option is that the manager responsible to create business value own the responsibility for performance evaluation. An additional option is that the individuals with an appropriate skill-set and experience execute performance evaluation as the only management activity.

The line management in an organization is the group of part-time managers executing performance evaluation with employees that act in different teams spread over the organization.

Trends in organizational development coming from different sources suggest this approach. Examples are the LeSS (Large Enterprise Scaled Scrum) framework or the squads and tribes organization as defined by Henrik Kniberg⁶ and implemented by Spotify⁷.

An over-arching observation concerning agile organizations is that line management (interpreted as the control of direct headcounts) loses importance and weight. The new type of manager focuses on the contribution to the company's vision and the responsibility for decisions that influence the future development of the company and organization.

⁶ **Henrik Kniberg** is an agile and lean coach at Crisp in Stockholm, working mostly with Spotify and Lego.

⁷ **Spotify** is a music, podcast, and video streaming service, officially launched on 7 October 2008. It was developed by startup Spotify AB in Stockholm, Sweden.

List of Acronyms Used in this Paper

<u>Acronym</u>	<u>Explanation</u>
CEO	Chief Executive Officer
CFO	Chief Financial Officer
CIO	Chief Information Officer
CoP	Community of Practice
LeSS	Large Enterprise Scaled Scrum
RUP	Rational Unified Process
SAFe	Scaled Agile Framework
SSADM	Structured Systems Analysis and Design Method

Acknowledgements

A sincere thank you to SyEN Editor Dr. Ralph Young for his extensive and very helpful collaboration with me.

References

The Scaled Agile Framework. www.scaledagileframework.com, copyright by Scaled Agile INC.

The Large Enterprise Scaled Scrum Framework. <https://less.works/>, copyright © 2014 ~ 2017 The LeSS Company B.V. All Rights Reserved.

Scaling Lean and Agile Development. Craig Larman and Bas Vodde. Addison Wesley. 8 Dec 2008. ISBN-13: 978-0321480965.

Scaling Agile @ Spotify, with Tribes, Squads, Chapters & Guilds by Henrik Kniberg & Anders Ivarsson. Oct 2012.

Spotify engineering culture, animated videos under <https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/> and <https://labs.spotify.com/2014/09/20/spotify-engineering-culture-part-2/>

Disciplined Agile Delivery. Scott Ambler. IBM Press, 23 May 2012. ISBN-13: 978-0132810135.

Management 3.0. Jurgen Appelo. Addison Wesley, 28 Dec 2010. ISBN-13: 978-0321712479.

Beyond Budgeting. Jeremy Hope, Robin Fraser, and Charles T. Horngren. Harvard Business Review Press, 1 Mar 2003. ISBN-13: 978-1578518661.

Leading Change. John P. Kotter. Harvard Business Review Press. ISBN-13: 978-0875847474.

Spiral Development: Experience, Principles, and Refinements. Spiral Development Workshop, February 9, 2000, Barry Boehm, edited by Wilfred J. Hansen, July 2000, SPECIAL REPORT CMU/SEI-2000-SR-008.

Vee Model, (German: V-Model XT) <http://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/2.1/Dokumentation/V-Modell-XT-HTML/index.html>, official website of the German reference model for system development projects, Version 2.1, 2015.

Organize for Complexity: How to Get Life Back Into Work to Build the High-Performance Organization. Niels Pflaeging. 19 March 2014, Betacodex Publishing. ISBN-13: 978-0991537600

Holacracy: The Revolutionary Management System that Abolishes Hierarchy. Brian J. Robertson.

2 June 2016, Portfolio Penguin. ISBN-13: 978-0241205860.

Leading when you're Not the Boss: How to Get Things Done in Complex Corporate Cultures. Roger Strathausen. 1st Edition. Apress. ISBN-13: 978-1484217474.

ARTICLE

Beyond the “Vee” Model: The Wedge Model™

by

Robert J. Halligan, CPEng FIE Aust

Chairman and Managing Director, Project Performance International

Chairman of the Board, Certification Training International

rhalligan@ppi-int.com

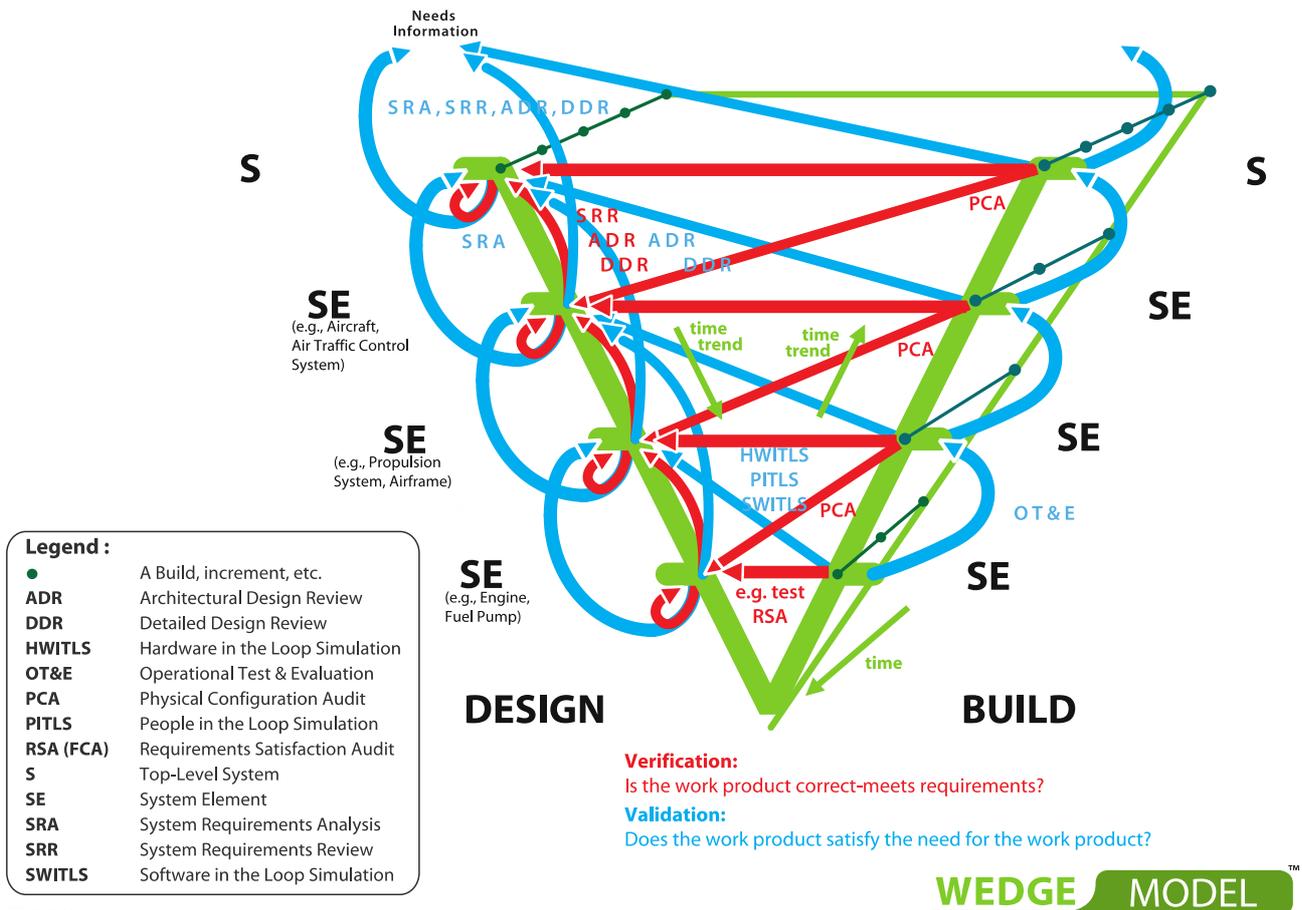
Abstract

The Vee Model, first defined by Rook in 1986, has served well as a depiction of design of non-trivial systems through physical levels of solution description on the left-hand side of the Vee, and physical realization of system elements, together with their integration to create higher physical level system elements, on the right-hand side of the Vee. The Vee model has commonly shown verification of system elements on the right-hand side, against requirements for each element on the left. Whilst valuable, this depiction is a very limited representation of the reality of development of non-trivial systems. The Wedge Model™ extends the basic Vee model to incorporate verification and validation of requirements, design, system elements and system, within a multiple-build development strategy.

Description

The Wedge Model™ is a derivative of the well-known Vee Model (Rook, 1986). The Wedge Model™ shows the products of design on the left-hand side of the wedge, and system integration on the right-hand side, with *verification* (is the work product correct with respect to the inputs used to create it - usually "meets requirements"?) and *validation* (does the work product correspond to the need for the work product?) superimposed. Colloquially, *verification* is - have we done the job right, and *validation* is - have we done the right job?

The Wedge Model™ shows multiple builds in the third dimension - a second time axis with time coming out of the page towards the reader. Thus, the face of the Wedge represents the last build (version, release, etc.) in each case. All of the *verification* and *validation* concerns on the face are replicated back onto the earlier builds. Note that not all potential *verifications* and *validations* will be carried out on all of the products associated with all of the builds. It is an engineering management role to decide which of the products will be subject to *verification* and *validation*, and to what degree.



A description of the model follows. Verification is shown in red. Validation is shown in blue. The subjects of verification and validation are at the arrow tails. The references for verification and validation are at the arrow heads.

1. It all starts with the capture and validation of requirements at the top-level, in this case the enterprise or business or capability system level. That validation is carried out using mainly the methods of System Requirements Analysis (SRA), the validation activity shown in blue at the top left.
2. People make mistakes in doing system requirements analysis. Therefore, we want to verify these work products of system requirements analysis. The most common and effective way of doing this requirements work product verification is with a system requirements review, shown in red. Other methods of verification of requirements analysis work products may also be employed.
3. System Requirements Reviews (SRRs) also provide some additional requirements validation, although they are not very effective for this purpose. SRR is also shown in blue at the top because of its requirements validation contribution.
4. At some point in time, requirements are formally released and drive design. Architecture (i.e. conceptual design) is produced, and there is a desire to verify architecture before committing to the much greater effort of detailed design. This purpose is most commonly served by an Architectural Design Review (ADR) shown in red. Architecture may also be verified by means additional to or alternative to design review. Here, the term architectural design is used with reference to a system of interest (Sol), and a physical level of design one physical level below the Sol.
5. It is all very well verifying architecture, but that requirements validation and design verification can never be perfect is well understood. So, the possibility exists that the design is still inconsistent with meeting needs. We are therefore interested in design validation. A low cost, and the most common, way of achieving this design validation is to have stakeholders participate in the design review, thereby identifying any failure to meet need. ADR is therefore also shown in blue for its validation contribution, but only if stakeholders participate.
6. If design is invalid for the usual reason that design is invalid, that is, because requirements are invalid, then we have achieved more requirements validation. Requirements validation is two steps removed from the primary purpose of design reviews, but design reviews *are* purposefully used to achieve additional requirements validation by inviting stakeholder participation.
7. The corresponding comments apply to Detailed Design Review (DDR), for design verification, design validation, and additional requirements validation, the last two only if stakeholders participate.
8. Eventually requirements on the sub-system, here for example the aircraft, are released, and it all starts again, and again, and again, until our development stops, here at, for example, the engine. Note that at level 2, where the aircraft appears in the example, another element will be the project system. If the Vee for that system is taken out and placed beside the aircraft Vee, we get the “Dual

Vee” (Forsberg and Mooz, 1992), a view that is useful where the emphasis is the management of a project to create a product. But we lose the enterprise system (etc.) and the references for validation of the product that the project is to deliver.

9. Eventually the engine comes into existence and there is a desire to verify that the engine meets the requirements for the engine, typically by test and other verification methods performed on the engine. Another verification method sometimes used is a Requirements Satisfaction Audit (RSA), also show in red. This is sometimes called a Functional Configuration Audit. The activity goes through the whole body of evidence to answer the question “can we reasonably conclude that the engine meets the requirements for the engine?”. That evidence examined includes test and other direct verification results, verification procedures used, test cases used, calibration of test equipment, system configuration data, the disposition of failures, and any evidence of falsification of test results.
10. Having verified that the engine meets its requirements, we are now concerned with validating the engine - does the engine satisfy the need for the engine? This is most commonly achieved by operational test and evaluation, for example flight test. Equivalent methods in other sectors are clinical trials in the medical sector, and test marketing in consumer products.
11. Flight testing an engine and finding that it crashes the aircraft is an undesirable approach. Another method of subsystem validation is Hardware in the Loop Simulation, where the engine is exercised in a partly physical and partly simulated environment, and the results of the simulation analyzed to draw a conclusion on meeting need. For space subsystems and systems, “in-the-loop simulation” may be the only viable means of subsystem or system validation.
12. Eventually, through system integration, the propulsion subsystem comes into existence. It is verified by testing etc. as meeting its requirements. Aggregates of system elements are created in a typical system integration activity. Each aggregate is usually subject to integration testing. Integration testing can be regarded as a small “v” verification activity, performed by or for the system developers and lacking the independence normally sought for a formal verification activity. Some small “v” validation of system elements is also achieved within system integration.
13. The red diagonal line pointing downwards indicates another verification activity. The activity addresses the concern “is the actual propulsion system built as described by its design description?”. If the answer is no, we are compromised in maintaining it, replicating it, adapting it and fixing latent defects. A Physical Configuration Audit (PCA) may be used for this purpose. The PCA painstakingly compares the actual product against its design description, and identifies any inconsistencies.
14. Verification and validation generally continue up to the top-level system. At the level of the aircraft in the example, validation would typically be in the form of an operational trial, while at the top level for a military capability system, validation would typically be via a military exercise.

15. However, reality is that systems and subsystems will often be developed in multiple builds (releases, versions), sometimes for end use and sometimes for purely engineering reasons, or technical management reasons, for example, risk reduction. The Wedge Model™ shows multiple builds in the third dimension - a second time axis with time coming out of the page towards the reader. Thus, the face of the Wedge represents the last build (version, release, etc.). Each build has requirements, design and implementation. All of the *verification* and *validation* concerns on the face of the wedge are replicated back onto the earlier builds. Note that not all potential *verifications* and *validations* will be carried out on all of the products associated with all of the builds. It is an engineering management role to decide which of the products will be subject to *verification* and *validation*, and to what degree.

The “Vee” model is sometimes misunderstood to be a development model, and a time-sequential Waterfall development model at that. This is not the case. The original “Vee” represents the end products on the design and build sides of the “Vee”, but says almost nothing about how the developer got there. The Wedge Model™ adds some of the story as to how the developer got there, but only insofar as intermediate design artifacts and corresponding intermediate builds. There are time trends, but no strict time axes, even for incremental and evolutionary builds. These could be sequential or partly concurrent on each side of the Wedge. The timing dependencies are finishing, not starting or doing (we have to have finished building the engine before we can finish building the propulsion system).

A “Project System” invariably appears at Level 2 in the Wedge Model™. The Project System has system elements, and lower physical level system elements, and lower again, and again, down to, for example, individual test procedures and emails. Each subject to the same principles verification and validation principles and relationships.

Conclusion

The Wedge Model™ provides a comprehensive representation of the end products and intermediate products within the development of larger, more complex systems, together with the superimposition of the subjects of verification and validation, the references for verification and validation, and representative methods of verification and validation, within both single build and multiple build development strategies for the system itself and for each developmental system element.

References

Forsberg, K., and Mooz, H. “The relationship of systems engineering to the project cycle”. *Engineering Management Journal* Vol. 4, No. 3, 1992, pp. 36-43.

Rook, Paul. “Controlling Software Projects” in *Software Engineering Journal* (Vol 1, No 1, pp. 7-10, January 1986, ISSN 0268-6961.

ARTICLE

Integrating Program Management and Systems Engineering

by

Ralph R. Young, Editor, SyEN

This Section in last month's SyEN provided a summary of the reporting we have done concerning a new book, *Integrating Program Management and Systems Engineering* (IPMSE), a collaboration of the International Council on Systems Engineering (INCOSE), the Project Management Institute (PMI), and the Consortium for Engineering Program Excellence (CEPE) at the Massachusetts (USA) Institute of Technology (MIT). In addition, we provided a summary of Chapter 1 of "The Book".

This month we provide a summary of Chapter 2, The Engineering Program Performance Challenge. This Chapter provides examples of engineering programs, both successful and not very successful. The Chapter concludes with a statement of today's engineering program performance challenge.

The London 2012 Olympic Committee adopted a management approach that demonstrated application of core systems engineering and program management principles with an emphasis on integration and collaboration, as follows:

- Clear vision/Goals and benefits: These were two key goals that governed every aspect of planning, execution, and management.
- Governance: The Olympic Delivery Authority (ODA) integrated the various governmental bodies and agencies that needed to be involved into a single interface with planning authority.
- Management Approach: Organizers maintained an integrated systems view over all aspects of the Games and leveraged program and project management to maintain alignment and integration.
- Integrated Planning and Risk Management: With a constant eye on the critical path, integrated planning incorporated dependency management, design management, physical integration, and scope change management.

- Customer and Stakeholder Management: The ODA maintained proactive customer and stakeholder engagement and communications at both the project and program levels.
- Scope and Requirements Control: Organizers established a change control system to manage requirements growth, risk, and necessary trade-offs to achieve program-level objectives. The program also incorporated verification and testing processes into the program schedule and provided sufficient time to address defects well in advance of the opening ceremonies.

The under-performance of the U.S. Department of Defense was astounding. The accumulated cost overrun of the largest 96 engineering programs had reached US\$300 billion when assessed in 2009. Engineering-related costs represented almost half of the expense increases associated with those programs. Most of those programs were at least two years late in their delivery. In project and program environments, complexity is a characteristic that is difficult to manage due to human behavior, system behavior, and ambiguity.

The experience of the Denver International Airport automated baggage handling system provides insight into challenges of integrated planning, stakeholder management, requirements management, and technical scalability. The program's projected cost in 1989 was close to US\$2billion; the program began in December 1989. Key stakeholders (major airlines) were not involved in the airport's initial design. A strong change control process for documenting, evaluating, considering alternatives, or prioritizing requirements changes was lacking. The baggage system was complicated and had to be designed around fixed building construction plans, which included several sharp, angled turns. There was a compressed timeline for planning, development, testing, and deployment. There was little risk management planning and mitigation. Engineers failed to consider important design elements for the airport's system. There was no back-up plan for baggage handling. The airport opened for business two years behind schedule.

The U.S. Army and its Future Combat Systems (FCS) Program evolved over a ten-year period before its cancellation in 2009. This program exemplifies many of the challenges and opportunities associated with engineering programs. It yielded some new best practices as well as many important lessons learned. The FCS program management approach attempted to revolutionize established program practices while remaining aligned with having "one team" and System of Systems (SoS) objectives. There were issues between the Army and the Lead Systems Integrator (LSI). Executives pushed for an overly ambitious schedule, reducing the six-year milestone to two years. The FCS consisted of 18 individual systems and required an over-arching network and operating system tying the components together. The rush to change major

requirements placed severe constraints on the program. Senior leaders did not authorize needed staff. Program requirements and priorities shifted significantly in response to external pressures. Army officials cancelled the program in 2009. The total cost of the program is estimated at about US\$20 billion, with very little resulting warfighter capability. FCS was a large enough part of the Army acquisition budget that it displaced and delayed other programs.

In the early 2000s, Volkswagen had ambitions to be the world's most profitable automotive company. To achieve that objective, the company had to increase its share of the U.S. market. The problem was that U.S. clean air standards targeted the type of pollution from oxides of nitrogen that diesel engines like those in Volkswagen vehicles produced. Unfortunately, VW's long-standing chief executive created a culture of fear that did not tolerate failure. Software engineers were tasked with developing a "defect device" that could sense emission testing conditions and put the vehicle into a safety mode in which the engine ran below normal power and performance. For six years, VW continue to produce and sell its high performing cars around the world. News reports indicated that an internal whistle blower notified a senior company official of the use of the defect device as early as 2011. Only after extensive testing and reporting of non-compliant emissions during the 2012-2015 timeframe did VW officials admit the cover up. Then the truth came out. VW's case reflects a system in which the failure of program leadership damaged the organization's brand and reputation globally.

Research performed to provide the knowledge base for IPMSE included *The Guide to Lean Enablers for Managing Engineering Programs*. The research uncovered the following major challenges that affect engineering program performance:

- Firefighting (reactive program execution).
- Unstable, unclear, and incomplete requirements.
- Mismanagement of program culture, team competency, and knowledge.
- Insufficient alignment and coordination of the extended enterprise.
- Unclear roles, responsibilities, and accountability.
- Insufficient program planning.
- Lack of proactive program risk management.
- Poor program acquisition and contracting practices.

It's clear that an improved approach is urgently needed. Current program management and systems engineering performance is not workable given that societies around the world require the benefits that only major engineering programs can deliver. Something must change.

SYSTEMS ENGINEERING NEWS

INCOSE Washington Metropolitan Area Chapter to Sponsor a STEM Initiative

Science, Technology, Engineering, and Mathematics (STEM) are at the cornerstone of being a consummate systems engineer and are vital to our future. In a world that is becoming increasingly complex, it is often through practicing STEM professionals that we are able to move the needle on invention and innovation for the better good of all society. All young people should be able to think critically and reason deeply so that they have the opportunity to become scientists, innovators, educators, and researchers solving current and to-be discovered global challenges. INCOSE's mission is to "share, promote and advance the best of systems engineering from across the globe for the benefit of humanity and the planet." The best way to share, promote, and advance the SE practice is through the enablement of our future leaders, equipping them today with the skills and ability to tackle our toughest global problems of tomorrow. INCOSE WMA will be standing up a STEM working group whose purpose is to address the very issues identified above and instill the strategic long term 'systems' thinking in our young leaders.

[More Information](#)

Software Brittleness May Harden Embedded Systems

Legacy systems in the government are hard to secure against cyberattack, and that goes double for the embedded, real-time systems at the heart of the many control systems in advanced weapons, power plants and mission-critical infrastructure.

Galois, a U.S. based secure software developer, has linked up with the Office of Naval Research under a three-year, \$2.7 million contract to tackle the problem. Leveraging what it calls "software brittleness," Galois aims to complement the fault-tolerance that already exists in these systems to boost their resilience in the event of attack.

The applications that run these systems already contain a certain inherent level of brittleness, said Tristan Ravitch, Galois' principal investigator on the project. It allows them to automatically fail over to a known-good state when problems arise, ensuring the systems continue to run. That's where the resilience comes in.

"We're trying to enhance that," he said. "We want such events [when an attack occurs] to actually turn into a crash, rather than have it go into some weird state," which can then be exploited by an attacker.

The brittleness that Galois wants to insert into the software would make the code crash faster, he said, before attacks, memory corruption and overflowing vulnerabilities can damage critical systems.

Galois' approach is to rewrite binary code, as the program is running, rather than the source code. It effectively takes a firmware image that it then runs through a tool that adds some capabilities to the binary, which is executed as a part of the code's normal runtime.

That gets around certain problems. When a compiler acts on code, it often makes decisions about what is and isn't valid, Ravitch said, and that doesn't leave much room for errors that attackers can exploit. Also, by acting on the binary code, the company's approach cuts out the problems in many regular "active" security solutions that rely on monitoring processes to make sure systems are doing what they are supposed to do.

[More Information](#)

INCOSE-PMI Collaboration

Collaboration between the Project Management Institute (PMI) and INCOSE has been steadily increasing over recent years. The extent of this collaboration is now evident with the announcement of a Program Management & Systems Engineering Virtual Conference to be held by PMI in partnership with INCOSE and MIT's Consortium for Engineering Program Excellence (CEPE) on 27 September, 2017. The slogan for the conference is:

"What happens when you integrate program management and systems engineering when managing complex engineering programs? Better outcomes."

The conference will take place on 27 September 2017 from 8:00am to 1:00pm USA ET. To register, click [here](#).

Nominations are Open for The Lt Gen Thomas R. Ferguson, Jr. Systems Engineering Excellence Award Program

The Ferguson awards will be presented at the Systems Engineering Conference, October 23-27 2017, Springfield, VA. These awards are given every year to both a deserving individual and group who have demonstrated outstanding achievement in the practical application of Systems Engineering.

How to Nominate: Information on how to nominate an individual or group may be found on the System Engineering Division [Award webpage](#).

Nominations must be received not later than October 1, 2017 to be considered.

Winners will have demonstrated outstanding achievement in the practical application of systems engineering principles; promotion of robust systems engineering principles throughout the organization; or

effective systems engineering process development. Their achievements should have clearly helped achieve one or more of the following:

- significant cost savings due to new or enhanced processes, procedures and/or concepts;
- increased mission capabilities; or
- substantially increased performance.

For additional information on nominations please contact Tina Fletcher at tfletcher@ndia.org.

Nominations are due by noon eastern time, October 1, 2017.

INCOSE Webinar:

Agile Systems & Processes 106: Risk Management and Mitigation

Presented by

Rick Dove

Abstract: To be effective, systems/processes have to mate well with their operational environments. Operational environments are not static, they react to disturbances and evolve with opportunity and risk. Inserting a system into an environment is a disturbance. Sustaining a system in a dynamic environment requires compatible evolution. The environment is the problem space the system will occupy. Understanding the requirements for a compatible-to-the-space solution is best done before system functional requirements shape an incompatible path. Given enough understanding about the problem, effective solution requirements and features becomes (almost) obvious. The problem shapes and constrains effective solution. But how do we characterize the environment as a dynamic problem space and develop solution-response requirements; and then, how do we structure a solution for risk-mitigating agility? This webinar introduces methods for dynamic problem-space characterization, and reviews companion methods for risk-mitigating solution-space agility.

Date: 20 September 2017

Time: 11:00am - 12:00pm (EDT)

[More information](#)

SERC Talks:

What are the Top Ten Software Security Flaws?

Presented by

Gary McGraw

Abstract: Software security defects come in two categories: bugs in the implementation and flaws in the design. In the commercial marketplace, much more attention has been paid to finding and fixing bugs than has been paid to finding and fixing flaws. That is because automatically identifying bugs is a much easier problem than identifying design flaws. The IEEE Center for Secure Design was founded to address this issue head on. My presentation will cover the IEEE CSD's first deliverable by introducing and discussing how to avoid the top ten software security flaws. The content was developed in concert with Twitter, Google, Cigital, HP, Sadosky Foundation of Argentina, George Washington University, Intel/McAfee, RSA, University of Washington, EMC, Harvard University, and Athens University of Economics and Business. During the talk, I will introduce and discuss how to avoid the top ten software security design flaws. It's important, of course, to know that these flaws account for half of the defects commonly encountered in software security. But more important still is learning how to avoid these problems when designing a new system or revisiting an existing system.

Date: 4 October 2017

Time: 1:00 pm (ET)

[More information](#)

FEATURED ORGANIZATIONS

The International Systems Engineering Network

The International Systems Engineering Network (ISEN) is an international networking group on Linked In comprised of over 16,000 members of organizations that promote and support systems engineering and for all professionals who want to share their knowledge and experience related to systems engineering.

[More Information](#)

CONFERENCES AND MEETINGS

For more information on systems engineering-related conferences and meetings, please proceed to [our website](#).

SOME SYSTEMS ENGINEERING-RELEVANT WEBSITES

MITRE Systems Engineering Resources

MITRE's knowledge-sharing culture extends beyond the corporation and their customers. They often collaborate with the community of practitioners to advance the discipline and practice of systems engineering and a number of relevant artefacts are freely available on their website to the systems engineering community.

<https://www.mitre.org/capabilities/systems-engineering/se-resources>

SYSTEMS ENGINEERING PUBLICATIONS

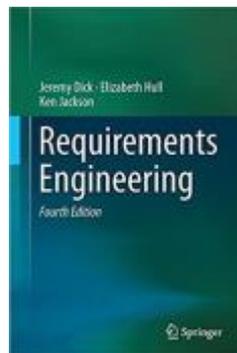
How Systems Engineering Can Help Fix Health Care

by

Alan Ravitz, Conrad Grant, and Peter Pronovost

Read the article [here](#).

Requirements Engineering



[Image source](#)

by

Jeremy Dick, Elizabeth Hull, and Ken Jackson

Book Description (from the Amazon web site):

Written for those who want to develop their knowledge of requirements engineering process, whether practitioners or students. Using the latest research and driven by practical experience from industry, **Requirements Engineering** gives useful hints to practitioners on how to write and structure

requirements. It explains the importance of systems engineering and the creation of effective solutions to problems. It describes the underlying representations used in system modeling and introduces the UML2, and considers the relationship between requirements and modeling. Covering a generic multi-layer requirements process, the book discusses the key elements of effective requirements management. The latest version of DOORS (Version 7) - a software tool which serves as an enabler of a requirements management process - is also introduced to the reader here.

[More Information](#)

“Suits you sir! – Choosing the Right Style of SE before Tailoring to Fit” Using Functional Failure Modes and Effects Analysis to Guide Selection of the Right Systems Approach

by

Duncan Kemp, Richard Beasley, and Samantha Williams

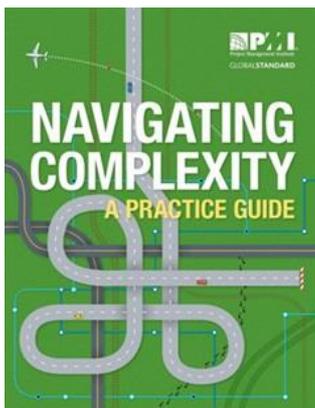
Despite significant evidence that Systems Engineering (SE) adds value, take-up is patchy. There is a growing volume of evidence that the 'understand-plan-do' paradigm behind conventional Systems Engineering only works in some circumstances. We have seen situations where SE has either over-complicated the solution or was unable to deliver quickly enough. Applying SE in these situations leads to both a specific failure *and* damages the credibility of SE more broadly. This paper:

- Proposes a new, two phase, tailoring process. Select an approach suitable for the situation then (if appropriate) decide what SE to do.
- Includes a Functional Failure Modes and Effects Analysis (FFMEA), which identifies functional failures modes, effects and mitigation activities applicable to the first stage of the tailoring process.
- Illustrates the FFMEA with seven case studies.

The proposed mitigations provide a high level blueprint for an enterprise that implements conventional SE, when, and only when, appropriate.

This paper was presented at the 25th Annual INCOSE International Symposium, Seattle, WA in July 2015. The paper is available to INCOSE members on the INCOSE web site [here](#).

Navigating Complexity: A Practice Guide



[Image source](#)

Book description from PMI web site:

In an increasingly volatile and uncertain world, most of today's CEOs expect to deal with a growing level of operational complexity. Executives, however, lack confidence that their organizations are equipped to handle those conditions effectively. *Navigating Complexity* builds upon substantial research reported in the *PMBOK® Guide* indicating that experienced project managers are more successful at completing highly complex projects. The practice guide expands on the *PMBOK® Guide* and PMI's other foundational standards and helps project managers to recognize complexity and cultivate the flexible mindset needed to cope effectively with change and unpredictability. The guide also helps practitioners apply the right tools and techniques to accomplish organizational goals.

[More Information](#)

Journal of Systems and Software on Quality Engineering and Management of Software-Intensive Systems (Special Issue)

The goal of this special issue is to collect current contributions relating to quality-engineering and management of software-intensive systems.

According to IEEE standards, software-intensive systems are described as "any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole" [IEEE Std 1471:2000] to encompass "individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, product families, whole enterprises, and other aggregations of interest". [IEEE Std 42010:2011]. Examples for software-intensive systems include embedded systems for avionics and automotive applications, large-scale heterogeneous systems, or business applications with special focus on web services. Software quality plays a pivotal role when developing and managing software-intensive systems.

The topics relevant to this special issue include, but are not restricted to, the following:

- Software engineering for embedded and cyber-physical systems
- Model-based development, components and services
- Software management
- Software process and product improvement
- Software product lines and software ecosystems
- Estimation and prediction in software and systems engineering
- Software engineering and technical debt
- Sustainable software engineering.

We seek high quality original submissions that have not been previously published and that are actually not under consideration for publication elsewhere as well as extended versions of selected papers of the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2017).

Preliminary dates:

- Submission: October 31, 2017
- First notification: December 15, 2017
- Revision: January 15, 2018
- Final decision: February 15, 2018
- Final manuscript: March 15, 2018

[More information](#)

SYSTEMS ENGINEERING TOOLS NEWS

Simulation Boom Stalls. CAE Experts Needed to Make Engineering Apps

It's widely accepted that Computer-aided engineering (CAE) technology will be useful to all industries, not just the traditional aerospace and automotive industries. But do they have the experts to make heads or tails of the technology?

CAE software continues to suffer from being the outsider at the party, despite predictions of its popularity. It's been well documented how simulation software can both innovate designs and increase product quality, while also reducing costs, risk and development time.

However, simulation is still not seeing the boom in users CAE vendors crave. What gives?

The pickle is that traditional CAE software is notoriously hard to use and the pool of simulation experts needed to run the software isn't meeting the demand.

Is the solution democratized simulation tools like engineering apps, templates, and fit for purpose CAE tools? Maybe, but the creation of these tools are themselves dependent on CAE experts. Even giving users access to simplified user interfaces (UI) in a Design Platform or Simulation in-CAE atmosphere requires that a Simulation expert eventually look things over.

[More Information](#)

Charles River Analytics Building Tools for Decision-making under Uncertain Conditions

Intelligent-systems developer Charles River Analytics has received a one-year contract extension -- awarded by the Defense Advanced Research Projects Agency (DARPA) and valued at close to \$500,000 -- for a system supporting Probabilistic Model-Based Programming Techniques for Prediction, Analysis, and Control, or PROMPT. The system, says the company, is intended for use by system engineers to reason under uncertain conditions.

Charles River Analytics says that PROMPT is designed to extend model-based systems engineering (MBSE) languages -- widely used in software design to add safe and effective performance features to mission-critical systems -- with probabilistic programming capabilities. According to the company, PROMPT enables engineers to link existing MBSE language elements to model-based probabilistic programming language constructs, which produces a probabilistic program that may be executed and analyzed, thereby allowing engineers to reason about the behaviors of systems operating under uncertainty, such as a minesweeping system that relies on noisy sensors for target identification.

Dr. Avi Pfeffer, chief scientist at Charles River Analytics, says of the effort: "We're helping engineers create system models to reason about the operating behavior of systems in uncertain environments. We're developing probabilistic extensions to the standard SysML systems modeling language, which we compile to our Figaro probabilistic programming language, allowing engineers to explore potential outcomes and tradeoffs through plugins to their systems engineering tools."

[More Information](#)

Dassault Systèmes and No Magic Establish Partnership to Boost Systems Engineering Solutions

[Dassault Systèmes](#) and [No Magic, Incorporated](#) have entered into a partnership to join their expertise in the Model-based systems engineering field. The two companies will integrate their industry solutions based on Dassault Systèmes' 3DEXPERIENCE platform addressing, among others, the need for

competitiveness in the development of connected and autonomous experiences in the aerospace and defense, transportation and mobility, high-tech, and life sciences industries.

Today's Internet of Things (IoT) has evolved into the "Internet of Experiences" where autonomous products and connected devices are integrating more and more software to digitally connect to the physical world around them, blending together to become part of a living experience shaped by interactions between products, nature and life.

To support these highly intricate and interconnected systems, industrial companies are looking for the ability to virtually co-design and simulate systems of systems, embedded systems, and software architectures across industries, support cross-disciplinary approaches, and shorten design and engineering innovation cycles through automation and systematic reuse of existing data.

With the 3DEXPERIENCE platform, customers can implement continuous 3D digital processes and address lifecycle aspects of an experience, including requirements, Systems of Systems Architecture Models, Systems and sub-Systems architecture, functional, logical, and physical 3D modeling simulation.

The 3DEXPERIENCE platform has been developed to support openness by providing the largest range of open standards and languages for System Engineering, such as STEP, Modelica, FMI, ReqIF, or OSLC. The result of the partnership will enrich it with the adoption of industry standard models and languages such as UML-SysML, UPDM, DoDAF, MODAF, or UAF.

[More Information](#)

Jama

Jama is a product development platform that enables collaboration across the development team and also integrates with a host of other development tools.

The tool capabilities include:

- Define, store and review requirements
- Share robust, contextual information and decisions
- Orchestrate work across multi-functional teams
- Collaboration
- Validation, Verification and Test
- Compliance
- Product Variants
- Workflow Governance

Jama provides integration with a large number of other tools:

- **JIRA:** Connecting with JIRA ensures consistency of data and end-to-end traceability from requirements to defects and tasks.
- **TFS:** Enhancing your teams' ability to collaborate, define and prioritize definitions into agile teams working in TFS.
- **Rally:** Seamlessly align customer requirements through to stories and tasks in agile-focused Rally environments.
- **HPQC:** Bring together business analysts with QA teams to ensure higher quality and traceability from start to finish.
- **VersionOne:** Keep Agile teams in VersionOne aligned with strategic product decisions to deliver better products and applications.
- **Enterprise Architect:** Engineers build system models and system requirements in Enterprise Architect. Jama provides cross-functional teams a place to openly discuss customer needs, manage changes, track decisions and capture approvals on project plans.
- **HPQC:** By integrating Jama with HP Quality Center (HPQC), teams can ensure 100% test coverage and eliminate unnecessary rework and defects that cost development teams 10x more to fix later in a release cycle.
- **MagicDraw**
- **Rally:** Jama's integration with Rally creates an ongoing feedback loop, giving everyone confidence that development activities map back to customer expectations and priorities.
- **Team Foundation Server (TFS):** Integrating Jama and TFS enables product teams and stakeholders working in Jama to collaboratively write, validate, approve and communicate requirements and story prioritization.
- **VersionOne:** Developers track active backlog items and sprints in VersionOne. Jama provides cross-functional teams a place to openly discuss customer needs, manage changes, track decisions and capture approvals on project plans.
- **TraceTronic:** TraceTronic's ECU-TEST integrates with Jama's Test Center to allow roundtrip syncing of test cases and results from automated testing of Electronic Control Units.
- **REST API:** Jama's REST API allows developers to build custom applications and integrations with other tools, extending the value of the Jama solution.

More information on Jama Software can be found [here](#).

EDUCATION AND ACADEMIA

Architecture and Systems Engineering: Models and Methods to Manage Complex Systems

Today's engineers face a critical challenge in designing, managing, and optimizing increasingly complex systems for the rapidly changing products of tomorrow. Employing industry case studies and the latest in systems thinking from MIT, this four-course online Professional Certificate program explores the various models and methods in systems engineering. You will learn quantitative methods in complex systems, Model-Based Systems Engineering (MBSE), and model management that will impact how you approach and solve problems.

[More Information](#)

INCOSE Institute for Technical Leadership

In 2015, INCOSE started an initiative to enhance and strengthen the Organization's Leadership by developing and providing a training program. The program's purpose is to accelerate the development of systems engineering leaders who will exemplify the best of the INCOSE organization and the systems engineering profession. Its primary benefits are the development of individual members as capable leaders, which benefits the involved members. INCOSE benefits include growing a pool of leaders, which can be used to help lead the Organization. Another intended outcome is to enhance INCOSE's International reputation in systems engineering.

To be selected for the program, INCOSE members must have first been nominated by an INCOSE board member or chapter president. Nominees are evaluated on their aptitude in both systems engineering and technical leadership, with evidence of personal accomplishments; comfort working in an uncertain world, and the willingness and ability to address problems at the socio-technical interface; interest in enhancing personal systems engineering leadership capabilities; and their potential to assume positions of leadership in the future.

The technical leadership institute is two-year program where new cohort is formed annually. There are four events per year: two face-to-face and two webcast. There are individual projects work between the events. Each cohort mentors the one following.

Please email info@incose.org for more information.

New Face, New Course Coming to Colorado State University Systems Engineering Program

One of the first steps of the systems engineering process is to determine what the customer's needs are. The Systems Engineering Program at Colorado State University (CSU) USA puts a strong emphasis on teaching students professional skills to succeed in industry and government, and to fulfill the workforce needs of regional partners, national labs and local companies. To continue to grow the Systems Engineering program and equip students with relevant skills, the program is welcoming a new faculty member this fall, along with a new course that has a strong focus on energy.

James Cale, who led the Distributed Energy Systems Integration Group at the National Renewable Energy Laboratory in Golden, Colorado, joined CSU as an associate professor in systems engineering on July 1. He will teach the program's new course, Coupled Electromechanical Systems, starting this fall. The course will center on the fundamental mechanics of energy conversion between electrical and mechanical systems. Cale brings to CSU over 10 years of industry and government experience, and firsthand knowledge of systems engineering problems in practice.

The Systems Engineering Program at CSU is known for its *versatile delivery model*, with instructors teaching courses delivered simultaneously on campus and online. This blended model provides increased access to students in the local community and at a distance, and is inspired by the university's land-grant heritage. "Continually adding breadth, depth, and capacity to the Systems Engineering Program is consistent with our strategic plan to meet the needs of the region, nation and planet," said Ron Segal, director of the Systems Engineering Program.

[More Information](#)

University of Alabama Huntsville (USA) Industrial and System Engineering Students Build Successful Business Model for Local Medical Practice

A group of Industrial and Systems Engineering and Engineering Management students at The University of Alabama in Huntsville (UAH) recently used simulation software to successfully improve a local medical practice.

"ISE (Industrial and Systems Engineering) students worked on the INNOVA Primary Care project as part of their senior design class. Their first semester on this project was in the spring, and they had two graduate students working with them to simulate and evaluate the current and new facility. ISE students used industrial engineering tools such as SIMIO simulation software, LEAN methods, and Six Sigma methodologies," said Dr. Sampson Gholston, UAH Assistant Professor of ISE and Engineering Management.

With the assistance of implemented analytical methods, UAH ISE students, patients and staff at INNOVA Primary Care (formerly Brooke MD Primary Care) successfully reached their model goal: improved patient care and satisfaction and facility efficiency. Dr. Brooke Uptagrafft runs the successful practice, a nationally-recognized Level III Patient Centered Medical Home (PCMH). Last year, the Huntsville/Madison County Chamber of Commerce named INNOVA as the 2016 Medical Practice of the Year. The primary care center recently expanded to a new facility near Crestwood Hospital.

LEAN methods implemented at the new clinic by UAH engineering students, assisted INNOVA in achieving special recognition as a “more highly efficient and patient-centered space.” The PCMH is a model or philosophy of primary care that is patient-centered, comprehensive, team-based, coordinated, accessible, and focused on quality and safety.

[More Information](#)

Collaborate Through Stories: Using Interactive Storytelling to Bridge the Gap between Engineers and Stakeholders

Typically, we make decisions after gathering all the necessary information and coming up with different options. We then choose a course of action that is likely to lead to the desired or best outcome.

For simple decisions, like choosing what to wear based on the weather forecast, picking the best option is fairly straightforward. However, for complex engineering problems such as smart manufacturing, production engineering, and real-time aircraft diversion planning, the considerations and possibilities are overwhelming. Furthermore, problems of this magnitude involve a large number of people from diverse backgrounds, such as systems engineering, finance, and program management, making communication and collaboration challenging.

Azad Madni, Executive Director of the Systems Architecting and Engineering Program and Professor of Astronautics at the Viterbi School of Engineering at the University of Southern California (USA), created an innovative solution that engages all involved parties, helps them develop a shared understanding of the problem, and allows them to collaboratively explore options with different assumptions before making informed decisions together. His method: interactive storytelling.

“Stories in literature are meant to engage you and evoke certain emotional responses from you,” said Madni. “Here, in complex systems engineering, stories are purposeful, interactive narratives constructed around the system being designed. Instead of a free-flowing adventure, stories in engineering strive to achieve certain desirable outcomes, in terms of desired system behaviors, with a minimum of unintended consequences. And we do that in the most engaging way possible to sustain team involvement.”

To create a story, Madni begins by using system models and data pertaining to each of the involved components to create and populate a virtual world. Take the example of an aircraft that needs to be rerouted to land on a different runway due to inclement weather. Using data about flights, weather and

ground personnel, as well as capabilities and constraints of the plane, runway and geographical location, a 3-D rendering of the system is brought to life in a virtual world using a video game engine.

Users can view and interact with the visual scenes, causing the story to unfold in particular ways. They can explore different scenarios based on their own what-ifs: What if the weather gets worse? What if that backup runway is unavailable? What if the airport at the destination is understaffed?

By making these worlds adaptable, stakeholders are able to steer the story through their interactions with the system, thereby cultivating a better understanding of the system's behaviors and limitations. After observing how each story plays out from the perspectives of the different stakeholders, they are better equipped to make informed, collective decisions in the real world.

[More Information](#)

STANDARDS AND GUIDES

International Standards Organization

The following standards are available for purchase from the International Office for Standardization:

- ISO/IEC 33001:2015 Information technology — Process assessment — Part 1: Concepts and terminology. <https://www.iso.org/standard/54175.html>
- ISO/IEC 33002:2015 Information technology — Process assessment — Requirements for Performing a Process Assessment. <https://www.iso.org/standard/54176.html>
- ISO/IEC TS 33030:2017 Information technology — Process assessment — an exemplar documented assessment process. <https://www.iso.org/standard/55121.html>

[More Information](#)

SOME DEFINITIONS TO CLOSE ON

A Glossary of Systems Engineering Terms from the Guide to the Systems Engineering Body of Knowledge (SEBoK) v. 1.8

SEBoK Glossary of Terms

http://sebokwiki.org/wiki/Category:Glossary_of_Terms

View acronyms and abbreviations used in the SEBoK

<http://sebokwiki.org/wiki/Acronyms>

The Internet of Things Explained

See http://www.webopedia.com/TERM/I/internet_of_things.html

PPI AND CTI NEWS

PPI's Systems Engineering Goldmine Hits 3.3GB

The Systems Engineering Goldmine, available to most SyEN subscribers, is an archive of downloadable documents on aspects of systems engineering. The archive includes guides, handbooks, papers, standards and many other items. Each document is indexed, with key words. A year ago the SEG stood at 1.7GB; this month we have hit 3.3GB. And there is more to come! We wish to acknowledge the sterling efforts of Wits University, South Africa, Masters engineering student and PPI Intern René King for this doubling of the SEG contribution to the systems engineering community. Take a bow, René.

Robert Halligan Presents for the INCOSE Singapore Chapter

PPI Managing Director, Mr. Robert Halligan delivered a presentation and discussion on “The Business Case for Systems Engineering for SMEs” to the INCOSE Singapore Chapter at a meeting hosted by NTUC (Singapore’s trade union). The Singapore Chapter has recently signed a MOU with NTUC to promote system engineering as a discipline/profession in Singapore. NTUC is keen on connecting to understand the discipline better and to provide training to industries beyond the traditional defence and aerospace sectors.

Leadership is Coming!

No, PPI has not been a dis-organized rabble devoid of leadership. Well, we don't think so!

But, Engineering Leadership training is coming. Four PPI personnel took part in a five-day PPI Engineering Leadership pilot course over 28 August to 1 September, 2017 - PPI Managing Director Robert Halligan, PPI Principal Consultant Alwyn Smit, PPI Professional Development Manager Suja Joseph-Malherbe, and PPI Engineering Intern René King. The pilot was conducted in the Devon Valley, near the gorgeous town of Stellenbosch, in the wine country of the Western Cape, South Africa. The course has been developed with long term collaborators Lynne Melis and Mary Gardner, outstanding professionals and two of the smartest people you will ever meet.

The impact - awe! To quote PPI Managing Director Robert Halligan: “*I wish I could have taken this course 50 years ago; I would have, over that period, been a substantially better engineer, better manager and better leader. To which I add, my quality of life would have been better, and my contribution to society greater!*”.

You may think that leadership is about CEOs. Sure, CEOs must be leaders, good leaders, great leaders. But leadership is pervasive. The best organisations are full of good/great leaders, from top to bottom. That is what this course is about. It is culture changing, it is life changing.

The course will be available for on-site delivery from December 2017 and for public delivery from March 2018. Please email contact@ppi-int.com for more information.

UPCOMING PPI AND CTI PARTICIPATION IN PROFESSIONAL CONFERENCES

PPI will be participating in the following upcoming events.

[NZDIA Forum 2017](#)

(Exhibiting)

10 - 11 October 2017

Wellington, New Zealand

[13th Annual INCOSE South Africa Conference](#)

(Exhibiting)

11 - 13 October 2017

Pretoria, South Africa

[11th Annual INCOSE Great Lakes Regional Conference \(GLRC11\)](#)

11 - 14 October 2017

Twin Cities, Minnesota, USA

[36th International Conference on Conceptual Modeling \(ER 2017\)](#)

6 - 9 November 2017

Valencia, Spain

[INCOSE UK, Annual Systems Engineering Conference \(ASEC\) 2017](#)

(Exhibiting)

21 - 22 November 2017

Warwick, UK

[8th CSD&M Paris](#)

12 - 13 December 2017

Paris, France

PPI AND CTI EVENTS

Systems Engineering Public 5-Day Courses

Upcoming locations include:

- Sydney, Australia
- Pretoria, South Africa
- Eindhoven, the Netherlands

Requirements Analysis and Specification Writing Public 5-Day Courses

Upcoming locations include:

- Las Vegas, NV, United States of America
- Amsterdam, the Netherlands
- Pretoria, South Africa

Systems Engineering Management Public 5-Day Courses

Upcoming locations include:

- Singapore
- London, United Kingdom
- Stellenbosch, South Africa

Requirements, OCD and CONOPS in Military Capability Development Public 5-Day Courses

Upcoming locations include:

- Amsterdam, the Netherlands
- Ankara, Turkey

Architectural Design 5-Day Course

Upcoming locations include:

- London, United Kingdom
- Pretoria, South Africa
- Stellenbosch, South Africa

Software Engineering 5-Day Courses

Upcoming locations include:

- Pretoria, South Africa
- London, United Kingdom
- Las Vegas, NV, United States of America

Human Systems Integration Public 5-Day Courses

We currently have no scheduled public courses. [Enquire](#) regarding an on-site proposal.

CSEP Preparation 5-Day Courses

(Presented by Certification Training International, a PPI company)

Upcoming locations include:

- Chantilly, VA, United States of America
- Madrid, Spain
- Pretoria, South Africa

Kind regards from the SyEN team:

Robert Halligan, Editor-in-Chief, email: rhalligan@ppi-int.com

Dr. Ralph Young, Editor, email: ryoung@ppi-int.com

Suja Joseph-Malherbe, Managing Editor, email: smalherbe@ppi-int.com

Project Performance International

2 Parkgate Drive, Ringwood North, Vic 3134 Australia Tel: +61 3 9876 7345 Fax: +61 3 9876 2664

Tel Brasil: +55 11 3958 8064

Tel UK: +44 20 3608 6754

Tel USA: +1 888 772 5174

Web: www.ppi-int.com

Email: contact@ppi-int.com

Copyright 2012-2017 Project Performance (Australia) Pty Ltd, trading as Project Performance International.

Tell us what you think of SyEN. Email us at syen@ppi-int.info.